



(11)

**EP 3 239 844 B1**

(12)

## EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention  
of the grant of the patent:  
**03.04.2019 Bulletin 2019/14**

(51) Int Cl.:  
**G06F 12/02<sup>(2006.01)</sup> G06F 12/12<sup>(2016.01)</sup>**

(21) Application number: **15877626.0**

(86) International application number:  
**PCT/CN2015/090765**

(22) Date of filing: **25.09.2015**

(87) International publication number:  
**WO 2016/112713 (21.07.2016 Gazette 2016/29)**

### (54) PROCESSING METHOD AND DEVICE FOR MEMORY PAGE IN MEMORY

VERARBEITUNGSVERFAHREN UND VORRICHTUNG FÜR SPEICHERSEITE IN EINEM  
SPEICHER

PROCÉDÉ ET DISPOSITIF DE TRAITEMENT POUR UNE PAGE DE MÉMOIRE DANS UNE  
MÉMOIRE

(84) Designated Contracting States:  
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB  
GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO  
PL PT RO RS SE SI SK SM TR**

(30) Priority: **15.01.2015 CN 201510021433**

(43) Date of publication of application:  
**01.11.2017 Bulletin 2017/44**

(73) Proprietor: **Huawei Technologies Co. Ltd.  
Shenzhen, Guangdong 518129 (CN)**

(72) Inventors:  
• **LIU, Duo**  
**Chongqing 400044 (CN)**  
• **SHAO, Zili**  
**Chongqing 400044 (CN)**

• **LONG, Linbo**  
**Chongqing 400044 (CN)**

(74) Representative: **Maiwald Patent- und  
Rechtsanwaltsgesellschaft mbH**  
**Elisenhof**  
**Elisenstraße 3**  
**80335 München (DE)**

(56) References cited:  
**WO-A1-2012/158514 WO-A2-2012/174216**  
**CN-A- 102 419 735 CN-A- 103 688 246**  
**CN-A- 103 914 410 US-A1- 2008 215 800**  
**US-A1- 2009 327 586 US-A1- 2012 030 405**  
**US-A1- 2012 297 121 US-A1- 2013 254 458**  
**US-A1- 2014 003 142 US-A1- 2014 003 142**

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

## Description

### TECHNICAL FIELD

[0001] The present invention relates to the field of storage technologies, and in particular, to a method and an apparatus for processing a memory page in memory.

### BACKGROUND

[0002] With development of virtualization technologies, requirements for hardware resources such as memory resources are increasingly high. For a conventional dynamic random access memory (Dynamic Random Access Memory, DRAM), energy consumption is high and capacity expansion is difficult. A new-type memory with low energy consumption, a large capacity, and non-volatility, such as a phase change memory (Phase Change Memory, PCM), may be expected to replace the conventional DRAM memory.

[0003] As a new generation non-volatile memory, a PCM uses reversible phase change of a material to store information to achieve single-level cell (Single-Level Cell, SLC) storage and multi-level cell (Multi-Level Cell, MLC) storage. An SLC features high costs, a small capacity, and a high speed, while an MLC features a large capacity, low costs, and a low speed.

[0004] Based on different features of the SLC and the MLC, if SLC storage is used, relatively high access performance may be obtained but a storage capacity may be restricted, and if is, there is still no effective solution that takes into account advantages of both the SLC and the MLC.

[0005] US 2012/030405 A1 describes an information processing device includes an OS and a virtual machine switching section. The OS accesses a hardware resource including a non-volatile semiconductor memory and a semiconductor memory used as a cache memory of the non-volatile semiconductor memory. The virtual machine switching section switches a virtual machine in exaction from a first virtual machine to a second virtual machine while a cache process is executed, when cache miss in a process executed by the first virtual machine is detected.

[0006] US 2014/003142 A1 describes a nonvolatile memory device comprising a first area of single-level cells (SLCs) and a second area of multi-level cells (MLCs). The device determines whether a free block can be created by copying data between memory blocks of the first area. If not, the device selects at least one memory block from the first area and allocates the selected memory block as free memory block by copying the data stored in the selected memory block of the first area to the second area.

### SUMMARY

[0007] The invention is set out in the appended inde-

pendent claims. Advantageous embodiments are defined by the appended dependent claims. The embodiments or examples of the following description which are not covered by the appended claims are considered as not being part of the invention according to this description.

### BRIEF DESCRIPTION OF DRAWINGS

10 [0008]

FIG. 1 is a schematic diagram of an application environment of embodiments of the present invention; FIG. 2 is a flowchart of a method for processing a memory page in memory provided in Embodiment 1 of the present invention;

FIG. 3 is a schematic diagram of conversion between an SLC and an MLC performed in Embodiment 1 of the present invention;

FIG. 4 is a flowchart of a method for processing a memory page in memory performed in Embodiment 2 of the present invention;

FIG. 5 is a schematic diagram of using a balloon technique (balloon technique) to balance memory allocation between VMs;

FIG. 6 is a schematic structural diagram of an apparatus for processing a memory page in memory provided in an embodiment of the present invention; and FIG. 7 is a schematic structural diagram of a device for processing a memory page in memory provided in an embodiment of the present invention.

### DESCRIPTION OF EMBODIMENTS

35 [0009] In the embodiments of the present invention, an idle memory page and an active memory page exist in a form of an SLC, and these memory pages are memory pages that may be accessed within a short time. Use of high-performance SLC storage mode for these memory pages can fully improve access performance. In addition, to take into account a storage capacity, when a quantity of available idle SLC memory pages in a virtual machine (Virtual Machine, VM) is less than a specified threshold, data in an inactive SLC memory page is converted to data in an MLC memory page for storage. Because a storage capacity of an MLC is twice that of an SLC, after data in two inactive SLC memory pages is converted to data in two MLC memory pages for storage, one idle SLC memory page may further be saved (one idle SLC memory page is converted to two MLC memory pages, and data in two inactive SLC memory pages is transferred to the two converted MLC memory pages, to obtain two idle SLC memory pages. That is, one idle SLC memory page is saved), thereby improving the storage capacity and reducing page faults.

[0010] As shown in FIG. 1, an application scenario of the embodiments of the present invention mainly includes a hardware layer, a virtualization management

layer, and a virtual machine layer. The hardware layer includes hardware devices such as a central processing unit (Central Processing Unit, CPU) supporting virtualization, a PCM-based main memory, and a flash (Flash)-based external storage. The virtual machine layer is responsible for running a mobile operating system. The virtualization management layer is mainly responsible for coordination work such as virtual memory scheduling, input and output (Input/Output, I/O) request scheduling, and CPU task scheduling. The embodiments of the present invention improve a memory scheduling algorithm in the virtualization management layer. In the memory scheduling, allocation of an SLC memory page and an MLC memory page is adjusted according to activity statuses of memory pages, and an idle memory page and an active memory page are allocated to be an SLC storage manner. When a quantity of idle SLC memory pages in a virtual machine VM is less than a specified threshold, conversion between the SLC memory page and the MLC memory page is started, and data in an inactive SLC memory page is converted to data in an MLC memory page for storage, so as to save an idle SLC memory page for storage of new data. As shown in FIG. 1, in the embodiments of the present invention, the virtualization management layer (Hypervisor) is added with functions of an application-based SLC/MLC allocator (ASPA Page Allocator), an SLC/MLC address conversion table (when conversion is performed between an SLC and an MLC, because physical address space is changed, the SLC/MLC address conversion table is needed to record a conversion relationship), ballooning for multiple VMs (Ballooning for Multiple VMs), and SLC/MLC conversion (see detailed descriptions in the following Embodiments 1 and 2).

**[0011]** The embodiments of the present invention will be further explained in detail with reference to the accompanying drawings.

#### Embodiment 1

**[0012]** In the embodiment, a memory page in memory includes an idle SLC memory page, an active SLC memory page, an inactive SLC memory page, and an MLC memory page, and the idle SLC memory page has no data, data stored in the active SLC memory page is data accessed within a specified time period, and data stored in the inactive SLC memory page and the MLC memory page is data not accessed within the specified time period.

**[0013]** The specified time period herein is determined based on memory utilization (including memory access frequency and memory usage). A higher memory access frequency or higher memory usage indicates a shorter specified time period; or a higher memory access frequency and higher memory usage indicate a shorter specified time period.

**[0014]** For the foregoing different types of memory pages, in specific implementation, conversion may be

performed in real time between different types of memory pages based on a memory page access situation. A specific process is: allocating an idle SLC memory page for a memory allocation request after the memory allocation request is received, and re-identifying the idle SLC memory page as an active SLC memory page; re-identifying an inactive SLC memory page as an active SLC memory page after an access request for the inactive SLC memory page is received; and for an active SLC memory page, if no access request for the memory page is received within the specified time period, re-identifying the active SLC memory page as an inactive SLC memory page. In specific implementation, a memory page linked list may be used to store addresses of different types of memory pages (see detailed descriptions in subsequent Embodiment 2).

**[0015]** On the basis hereof, a process procedure shown in FIG. 2 may be executed, and includes the following steps.

**[0016]** S201: Convert one idle SLC memory page to two MLC memory pages when determining that a quantity of idle SLC memory pages in any VM is less than a specified threshold.

**[0017]** Herein, the specified threshold is determined based on a page fault probability and access performance. A smaller specified threshold indicates a higher probability of generating a page fault and higher memory access performance (because of a relatively small quantity of converted MLC memory pages). When the specified threshold is larger, the probability of generating a page fault is lower and the memory access performance is lower (because of a relatively large quantity of converted MLC memory pages). However, the page fault probability is not increased linearly with an increase of the specified threshold; accordingly, the access performance is not decreased linearly with the increase of the specified threshold. Therefore, a value that makes a page fault probability low and access performance relatively high may be obtained by using experimental tests in specific implementation.

**[0018]** In specific implementation, a given amount of memory is applied for each VM. After data is written to an idle SLC memory page in the memory, the memory page becomes an active SLC memory page which may be frequently accessed (read or written) and then enters progressively into an inactive period, and is almost hardly accessed, that is, becomes an inactive SLC memory page. In this embodiment of the present invention, idle SLC memory pages are constantly consumed for storing new data, and that memory resources are not released in a timely manner when VMs are busy with service may result in increasingly fewer SLC memory pages. When it is monitored that a quantity of idle SLC memory pages is less than the specified threshold, conversion between an idle SLC memory page and an MLC memory page is started.

**[0019]** S202: Copy data in two inactive SLC memory pages to the two converted MLC memory pages.

**[0020]** S203: Release storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages.

**[0021]** Referring to FIG. 3, FIG. 3 is a schematic diagram of conversion between an SLC and an MLC performed in Embodiment 1 of the present invention. Firstly, as described in S201, one idle SLC memory page is converted to two MLC memory pages (idle MLC memory pages at this moment). That is, each idle SLC memory page may be converted to two MLC memory pages (which is determined by physical properties of the SLC and the MLC), and storage space doubles.

**[0022]** Then, as described in S202, data in two inactive SLC memory pages is copied to the two converted MLC memory pages (the idle MLC memory pages become inactive MLC memory pages at this moment). Herein, for either the SLC memory page or the MLC memory page, an amount of data that can be stored in each memory page is the same; therefore data in every two SLC memory pages also needs to occupy two MLC memory pages.

**[0023]** Lastly, as described in S203, storage space of the two inactive SLC memory pages is released to be converted to two idle SLC memory pages. At the moment, the one idle SLC memory page initially converted to the two MLC memory pages is subtracted, that is, one idle SLC memory page is saved. The one idle SLC memory page may continue to be allocated for storing new data.

**[0024]** According to the foregoing Embodiment 1, an idle memory page to be accessed within a short time and an active memory page to be accessed frequently are planned to be in a high performance SLC storage mode, to fully improve memory access performance. In addition, when a quantity of available idle SLC memory pages in a VM is less than a specified threshold, data in an inactive SLC memory page is converted to data in an MLC memory page for storage, idle memory space may further be saved, thereby improving a storage capacity and reducing page faults.

**[0025]** In specific implementation, when a system works with a heavy load, especially working with overload, it may be impossible to find at least two inactive SLC memory pages, then data in two MLC memory pages may be directly written back to an external storage, and the two MLC memory pages are converted to one idle SLC memory page. Such conversion manner may lead to a page fault. However, data in the MLC memory pages is actually data in the inactive SLC and is not accessed frequently. This brings a relatively low extra overhead for the system. When the system workload is heavier and even an available MLC memory page cannot be found, it means that almost all memory pages are used in the SLC storage mode and are actively accessed. In this case, a super management program of a virtual platform may be started to balance memory allocation between different VMs. For detailed description, see following Embodiment 2.

## Embodiment 2

**[0026]** Referring to FIG. 4, FIG. 4 is a flowchart of a method for processing a memory page performed in Embodiment 2 of the present invention. The method includes the following steps.

**[0027]** S401: Establish different memory page linked lists to record activity statuses of memory pages: establish an idle SLC memory page linked list, an active SLC memory page linked list, an inactive SLC memory page linked list, and an MLC memory page linked list for each VM.

**[0028]** Herein, the idle SLC memory page linked list is used to store addresses of idle SLC memory pages, the active SLC memory page linked list is used to store addresses of active memory pages, the inactive SLC memory page linked list is used to store addresses of inactive memory pages, and the MLC memory page linked list is used to store addresses of MLC memory pages. Activity statuses of memory pages are recorded by using these memory page linked lists.

**[0029]** In specific implementation, the idle SLC memory page is a memory page that has no storage data, the active SLC memory page is used to store data frequently accessed recently, and the inactive SLC memory page and the MLC memory pages are used to store data not accessed within a specified time period. After a memory allocation request is received, an idle SLC memory page is allocated for the memory allocation request, and an address of the idle SLC memory page is transferred from the established idle SLC memory page linked list to the established active SLC memory page linked list. After an access request for an inactive SLC memory page is received, an address of the inactive SLC memory page is transferred from the established inactive SLC memory page linked list to the established active SLC memory page linked list. For the active SLC memory page, if no access request for the memory page is received within the specified time period, an address of the active SLC memory page is transferred from the established active SLC memory page linked list to the established inactive SLC memory page linked list.

**[0030]** S402: When a quantity of idle SLC memory pages in any VM is less than a specified threshold, adjust an allocation proportion between an SLC and an MLC in memory allocated to the VM: if a quantity of current inactive SLC memory pages is greater than or equal to 2, convert one idle SLC memory page to two MLC memory pages, and copy data in two inactive SLC memory pages to the two converted MLC memory pages; and release storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages; if a quantity of current inactive SLC memory pages is less than 2, write data in two MLC memory pages back to an external storage, and convert the two MLC memory pages to one idle SLC memory page.

**[0031]** S403: Balance memory allocation between VMs: if a quantity of current inactive SLC memory pages

is less than 2 and a quantity of current MLC memory pages is less than 2, obtain a memory resource of another VM for the any VM.

**[0032]** On a mobile virtualization platform, VMs running different applications have different memory requests based on requirements of different applications. When memory pages in a VM are almost in an active state, there is a large quantity of idle memory pages in another VM. By invoking a super management program of the virtual platform, part of memory is retrieved from the VM with a large quantity of idle memory pages (for example, the quantity of idle memory pages is greater than a preset value) and then is allocated to the VM with fewer idle memory pages currently.

**[0033]** As shown in FIG. 5, a balloon technique (balloon technique) is used to balance memory allocation between VMs. When memory allocated to a VM is retrieved by a super management program, a balloon drive program of the VM is expanded, and an idle SLC memory page may be released by using a balloon drive and is allocated to a VM with insufficient memory.

**[0034]** Based on the foregoing descriptions, the solution of the present invention has the following advantageous effects:

1) A high performance characteristic of an SLC is fully utilized, reducing a memory access delay. In this embodiment of the present invention, high-performance SLC memory is allocated for a new memory page request and a memory page written frequently, thereby improving access performance.

2) A large capacity characteristic of an MLC is fully utilized, expanding a memory capacity and reducing page faults. In this embodiment of the present invention, an inactive SLC memory page is converted to large-capacity MLC pages in a case of memory is insufficient, thereby increasing a memory capacity and reducing page faults.

3) System overall performance is improved: The overall performance of a system depends on three factors: a quantity of SLC access times, a quantity of MLC access times, and a quantity of page fault times. In this embodiment of the present invention, memory pages accessed frequently are planned to be low-delay SLC memory pages, and memory pages accessed less frequently are planned to be large-capacity MLC memory pages, thereby reducing a memory page access delay on the one hand and expanding a memory capacity, reducing page faults, and improving the overall performance of the system on the other hand.

4) Overall performance of multiple virtual machines VMs is improved: In this embodiment of the present invention, when available idle memory of a VM is insufficient, the super management program is invoked to retrieve an idle memory page in another VM and allocate the idle memory page to the VM with insufficient memory, thereby improving the

overall performance of the multiple VMs.

5) Storage and performance overheads are low: A system overhead in this embodiment of the present invention mainly includes a space overhead on an SLC/MLC address conversion table and a run-time overhead during an SLC/MLC conversion process, and these overheads are small; a statistical result indicates that the address conversion table occupies only memory about 0.2%, the SLC/MLC conversion process involves only a memory-to-memory operation, and the system overhead is very small.

**[0035]** According to a same inventive concept, an embodiment of the present invention further provides an apparatus for processing a memory page in memory corresponding to the method for processing a memory page in memory. A principle of the apparatus to solve a problem is the same as that of the method for processing a memory page in memory in the embodiment of the present invention, and for implementation of the apparatus, reference may be made to implementation of the method, and to avoid repetition, details may not be described herein.

**[0036]** Referring to FIG. 6, FIG. 6 is a schematic structural diagram of an apparatus for processing a memory page in memory provided in an embodiment of the present invention. A memory page in memory includes an idle single-level cell SLC memory page, an active SLC memory page, an inactive SLC memory page, and a multi-level cell MLC memory page. The idle SLC memory page has no data, data stored in the active SLC memory page is data accessed within a specified time period, and data stored in the inactive SLC memory page and the MLC memory page is data not accessed within the specified time period. The apparatus includes:

a conversion module 61, configured to convert one idle SLC memory page to two MLC memory pages when determining that a quantity of idle single-level cell SLC memory pages in any virtual machine VM is less than a specified threshold;

a copy module 62, configured to copy data in two inactive SLC memory pages to the two converted MLC memory pages; and

a release module 63, configured to release storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages.

**[0037]** Optionally, a higher memory access frequency or higher memory usage indicates a shorter specified time period; or a higher memory access frequency and higher memory usage indicate a shorter specified time period.

**[0038]** Optionally, a smaller specified threshold indicates higher memory access performance and a higher probability of generating a page fault.

**[0039]** Optionally, the conversion module 61 is further configured to, after determining that a quantity of idle

SLC memory pages in any VM is less than the specified threshold, if a quantity of current inactive SLC memory pages is less than 2, write data in the two MLC memory pages back to an external storage, and convert the two MLC memory pages to one idle SLC memory page.

**[0040]** Optionally, the conversion module 61 is further configured to, after determining that a quantity of idle SLC memory pages in any VM is less than the specified threshold, if a quantity of current inactive SLC memory pages is less than 2 and a quantity of current MLC memory pages is less than 2, obtain an idle memory page of another VM for the any VM.

**[0041]** Optionally, the apparatus further includes: a first allocation module 64, configured to allocate an idle SLC memory page for a memory allocation request after the memory allocation request is received, and re-identify the idle SLC memory page as an active SLC memory page.

**[0042]** Optionally, the first allocation module 64 is specifically configured to transfer an address of the idle SLC memory page from an established idle SLC memory page linked list to an established active SLC memory page linked list.

**[0043]** Optionally, the apparatus further includes: a second allocation module 65, configured to re-identify an inactive SLC memory page as an active SLC memory page after an access request for the inactive SLC memory page is received.

**[0044]** Optionally, the second allocation module 65 is specifically configured to transfer an address of the inactive SLC memory page from an established inactive SLC memory page linked list to the established active SLC memory page linked list.

**[0045]** Optionally, the apparatus further includes: a third allocation module 66, configured to, for an active SLC memory page, if no access request for the memory page is received within the specified time period, re-identify the active SLC memory page as an inactive SLC memory page.

**[0046]** Optionally, the third allocation module 66 is specifically configured to: transfer an address of the active SLC memory page from the established active SLC memory page linked list to an established inactive SLC memory page linked list.

**[0047]** Referring to FIG. 7, FIG. 7 is a schematic structural diagram of a device 70 for processing a memory page in memory provided in an embodiment of the present invention. The device 70 includes a processor 71, a storage 72, and a bus 73. The storage 72 includes memory 72a, a memory page in the memory 72a includes an idle SLC memory page, an active SLC memory page, an inactive SLC memory page, and an MLC memory page. The idle SLC memory page has no data, data stored in the active SLC memory page is data accessed within a specified time period, and data stored in the inactive SLC memory page and the MLC memory page is data not accessed within the specified time period. The storage 72 stores execution instructions, and when the

processing device 70 is running, the processor 71 communicates with the storage 72 by using the bus 73, so that the processor 71 performs the following execution instructions:

converting one idle SLC memory page to two MLC memory pages when determining that a quantity of idle single-level cell SLC memory pages in any virtual machine VM is less than a specified threshold; copying data in two inactive SLC memory pages to the two converted MLC memory pages; and releasing storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages.

**[0048]** Optionally, a higher memory access frequency or higher memory usage indicates a shorter specified time period; or a higher memory access frequency and higher memory usage indicate a shorter specified time period.

**[0049]** Optionally, a smaller specified threshold indicates higher memory access performance and a higher probability of generating a page fault.

**[0050]** Optionally, after the determining that a quantity of idle SLC memory pages in any VM is less than a specified threshold, the execution instructions performed by the processor 71 further include:

if a quantity of current inactive SLC memory pages is less than 2, writing data in the two MLC memory pages back to an external storage, and converting the two MLC memory pages to one idle SLC memory page, where M is a positive integer.

**[0051]** Optionally, after the determining that a quantity of idle SLC memory pages in any VM is less than a specified threshold, the execution instructions performed by the processor 71 further include:

if a quantity of current inactive SLC memory pages is less than 2 and a quantity of current MLC memory pages is less than 2, obtaining a memory resource of another VM for the any VM.

**[0052]** Optionally, the execution instructions performed by the processor 71 further include:

allocating an idle SLC memory page for a memory allocation request after the memory allocation request is received, and re-identifying the idle SLC memory page as an active SLC memory page.

**[0053]** Optionally, in the execution instructions performed by the processor 71, the re-identifying the idle SLC memory page as an active SLC memory page includes: transferring an address of the idle SLC memory page from an established idle SLC memory page linked list to an established active SLC memory page linked list.

**[0054]** Optionally, the execution instructions performed by the processor 71 further include:

re-identifying an inactive SLC memory page as an active SLC memory page after an access request for the inactive SLC memory page is received.

**[0055]** Optionally, in the execution instructions per-

formed by the processor 71, the re-identifying an inactive SLC memory page as an active SLC memory page includes: transferring an address of the inactive SLC memory page from an established inactive SLC memory page linked list to the established active SLC memory page linked list.

[0056] Optionally, the execution instructions performed by the processor 71 further include:

for an active SLC memory page, if no access request for the memory page is received within the specified time period, re-identifying the active SLC memory page as an inactive SLC memory page.

[0057] Optionally, in the execution instructions performed by the processor 71, the re-identifying the active SLC memory page as an inactive SLC memory page includes:

transferring an address of the active SLC memory page from the established active SLC memory page linked list to an established inactive SLC memory page linked list.

[0058] Persons skilled in the art should understand that the embodiments of the present invention may be provided as a method, a system, or a computer program product. Therefore, the present invention may use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware. Moreover, the present invention may use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, CD-ROM, an optical memory, and the like) that include computer-usable program code.

[0059] The present invention is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of this application. It should be understood that computer program instructions may be used to implement each process and/or each block in the flowcharts and/or the block diagrams and a combination of a process and/or a block in the flowcharts and/or the block diagrams. These computer program instructions may be provided for a general-purpose computer, a dedicated computer, an embedded processor, or a processor of any other programmable data processing device to generate a machine, so that the instructions executed by a computer or a processor of any other programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0060] These computer program instructions may also be stored in a computer readable memory that can instruct the computer or any other programmable data processing device to work in a specific manner, so that the instructions stored in the computer readable memory generate an artifact that includes an instruction apparatus. The instruction apparatus implements a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0061] These computer program instructions may also be loaded onto a computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or the another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0062] Although some preferred embodiments of the present invention have been described, persons skilled in the art can make changes and modifications to these embodiments once they learn the basic inventive concept. Therefore, the following claims are intended to be construed as to cover the preferred embodiments and all changes and modifications falling within the scope of the present invention.

## Claims

1. A method for processing a memory page in memory (72a), **characterized in that** the memory page in the memory (72a) comprises an idle single-level cell, SLC, memory page, an active SLC memory page, an inactive SLC memory page, and a multi-level cell, MLC, memory page, wherein the idle SLC memory page has no data, data stored in the active SLC memory page is data accessed within a specified time period, and data stored in the inactive SLC memory page and the MLC memory page is data not accessed within the specified time period; and the processing method comprises:

converting (S201) one idle SLC memory page to two MLC memory pages when determining that a quantity of idle single-level cell, SLC, memory pages in any virtual machine, VM, is less than a specified threshold;  
copying (S202) data in two inactive SLC memory pages to the two converted MLC memory pages; and  
releasing (S203) storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages.

2. The method according to claim 1, wherein a higher memory access frequency or higher memory usage indicates a shorter specified time period; or a higher memory access frequency and higher memory usage indicate a shorter specified time period.
3. The method according to claim 1 or 2, wherein a smaller specified threshold indicates higher memory access performance and a higher probability of generating a page fault.

4. The method according to any one of claims 1 to 3, after the determining that a quantity of idle SLC memory pages in any VM is less than a specified threshold, further comprising:  
if a quantity of current inactive SLC memory pages is less than 2, writing data in the two MLC memory pages back to an external storage, and converting the two MLC memory pages to one idle SLC memory page.
5. The method according to claim 4, after the determining that a quantity of idle SLC memory pages in any VM is less than a specified threshold, further comprising:  
if a quantity of current inactive SLC memory pages is less than 2 and a quantity of current MLC memory pages is less than 2, obtaining an idle memory page of another VM for the any VM.
6. The method according to any one of claims 1 to 5, wherein the method further comprises:  
allocating an idle SLC memory page for a memory allocation request after the memory allocation request is received, and re-identifying the idle SLC memory page as an active SLC memory page.
7. The method according to claim 6, wherein the re-identifying the idle SLC memory page as an active SLC memory page comprises: transferring an address of the idle SLC memory page from an established idle SLC memory page linked list to an established active SLC memory page linked list.
8. The method according to any one of claims 1 to 7, wherein the method further comprises:  
re-identifying an inactive SLC memory page as an active SLC memory page after an access request for the inactive SLC memory page is received.
9. The method according to claim 8, wherein the re-identifying an inactive SLC memory page as an active SLC memory page comprises: transferring an address of the inactive SLC memory page from an established inactive SLC memory page linked list to the established active SLC memory page linked list.
10. The method according to any one of claims 1 to 9, wherein the method further comprises:  
for an active SLC memory page, if no access request for the memory page is received within the specified time period, re-identifying the active SLC memory page as an inactive SLC memory page.
11. The method according to claim 10, wherein the re-identifying the active SLC memory page as an inactive SLC memory page comprises:  
transferring an address of the active SLC memory page from the established active SLC memory page linked list to an established inactive SLC memory page linked list.
12. An apparatus (70) for processing a memory page in memory (72a), **characterized in that** the memory page in the memory (72a) comprises an idle single-level cell, SLC, memory page, an active SLC memory page, an inactive SLC memory page, and a multi-level cell, MLC, memory page, wherein the idle SLC memory page has no data, data stored in the active SLC memory page is data accessed within a specified time period, and data stored in the inactive SLC memory page and the MLC memory page is data not accessed within the specified time period; and the apparatus comprises:  
  
a conversion module (61), configured to convert one idle SLC memory page to two MLC memory pages when determining that a quantity of idle single-level cell, SLC, memory pages in any virtual machine, VM, is less than a specified threshold;  
a copy module (62), configured to copy data in two inactive SLC memory pages to the two converted MLC memory pages; and  
a release module (63), configured to release storage space of the two inactive SLC memory pages to obtain two idle SLC memory pages.
13. The apparatus according to claim 12, wherein a higher memory access frequency or higher memory usage indicates a shorter specified time period; or a higher memory access frequency and higher memory usage indicate a shorter specified time period.
14. The apparatus according to claim 12 or 13, wherein a smaller specified threshold indicates higher memory access performance and a higher probability of generating a page fault.
15. The apparatus according to claim 14, wherein the conversion module (61) is further configured to, after determining that a quantity of idle SLC memory pages in any VM is less than the specified threshold, if a quantity of current inactive SLC memory pages is less than 2, write data in the two MLC memory pages back to an external storage, and convert the two MLC memory pages to one idle SLC memory page.
16. The apparatus according to claim 15, wherein the conversion module (61) is further configured to, after determining the quantity of idle SLC memory pages in any VM is less than the specified threshold, if a quantity of current inactive SLC memory pages is less than 2 and a quantity of current MLC memory pages is less than 2, obtain an idle memory page of another VMs for the any VM.

17. The apparatus according to any one of claims 12 to 16, wherein the apparatus further comprises: a first allocation module (64), configured to allocate an idle SLC memory page for a memory allocation request after the memory allocation request is received, and re-identify the idle SLC memory page as an active SLC memory page. 5
18. The apparatus according to claim 17, wherein the first allocation module (64) is specifically configured to transfer an address of the idle SLC memory page from an established idle SLC memory page linked list to an established active SLC memory page linked list. 10
19. The apparatus according to any one of claims 12 to 18, wherein the apparatus further comprises: a second allocation module (65), configured to re-identify an inactive SLC memory page as an active SLC memory page after an access request for the inactive SLC memory page is received. 15 20
20. The apparatus according to claim 19, wherein the second allocation module (65) is specifically configured to transfer an address of the inactive SLC memory page from an established inactive SLC memory page linked list to the established active SLC memory page linked list. 25
21. The apparatus according to any one of claims 12 to 20, wherein the apparatus further comprises: a third allocation module (66), configured to, for an active SLC memory page, if no access request for the memory page is received within the specified time period, re-identify the active SLC memory page as an inactive SLC memory page. 30 35
22. The apparatus according to claim 21, wherein the third allocation module (66) is specifically configured to: transfer an address of the active SLC memory page from the established active SLC memory page linked list to an established inactive SLC memory page linked list. 40 45

## Patentansprüche

1. Verfahren, um eine Speicherseite in einem Speicher (72a) zu verarbeiten, **dadurch gekennzeichnet, dass** die Speicherseite im Speicher (72a) eine untätige Speicherseite mit einstufigen Zellen, SLC-Speicherseite, eine aktive SLC-Speicherseite, eine inaktive SLC-Speicherseite und eine Speicherseite mit mehrstufigen Zellen, MLC-Speicherseite, umfasst, wobei die untätige SLC-Speicherseite keine Daten enthält, Daten, die in der aktiven SLC-Speicherseite gespeichert sind, Daten sind, auf die in ei-

nem bestimmten Zeitraum zugegriffen wird, und Daten, die in der inaktiven SLC-Speicherseite und der MLC-Speicherseite gespeichert sind, Daten sind, auf die im bestimmten Zeitraum nicht zugegriffen wird; und das Verarbeitungsverfahren Folgendes umfasst:

Umwandeln (S201) einer untätigen SLC-Speicherseite in zwei MLC-Speicherseiten, wenn bestimmt wird, dass eine Anzahl von untätigen Speicherseiten mit einstufigen Zellen, SLC-Speicherseiten, in einer beliebigen virtuellen Maschine, VM, geringer als ein bestimmter Schwellenwert ist;

Kopieren (S202) von Daten in zwei inaktiven SLC-Speicherseiten zu den zwei umgewandelten MLC-Speicherseiten; und

Freigeben (S203) des Speicherplatzes der zwei inaktiven SLC-Speicherseiten, um zwei untätige SLC-Speicherseiten zu erhalten.

2. Verfahren nach Anspruch 1, wobei eine höhere Speichertzugriffsfrequenz oder eine höhere Speicherverwendung einen kürzeren bestimmten Zeitraum anzeigt oder eine höhere Speichertzugriffsfrequenz und eine höhere Speicherverwendung einen kürzeren bestimmten Zeitraum anzeigt.
3. Verfahren nach Anspruch 1 oder 2, wobei ein kleinerer bestimmter Schwellenwert eine höhere Speichertzugriffsleistung und eine höhere Wahrscheinlichkeit des Erzeugens eines Seitenfehlers anzeigt.
4. Verfahren nach einem der Ansprüche 1 bis 3, das nach dem Bestimmen, dass eine Anzahl von untätigen SLC-Speicherseiten in einer beliebigen VM geringer als ein bestimmter Schwellenwert ist, ferner Folgendes umfasst:  
dann, wenn eine Anzahl von aktuell inaktiven SLC-Speicherseiten geringer als 2 ist, Schreiben von Daten in den zwei MLC-Speicherseiten zurück zu einem externen Speicher und Umwandeln der zwei MLC-Speicherseiten zu einer untätigen SLC-Speicherseite.
5. Verfahren nach Anspruch 4, das nach dem Bestimmen, dass eine Anzahl von untätigen SLC-Speicherseiten in einer beliebigen VM geringer als ein bestimmter Schwellenwert ist, ferner Folgendes umfasst:  
dann, wenn eine Anzahl von aktuell inaktiven SLC-Speicherseiten geringer als 2 ist und eine Anzahl von aktuellen MLC-Speicherseiten geringer als 2 ist, Erhalten einer untätigen Speicherseite von einer weiteren VM für die beliebige VM.
6. Verfahren nach einem der Ansprüche 1 bis 5, wobei das Verfahren ferner Folgendes umfasst:

- Zuordnen einer untätigen SLC-Speicherseite für eine Speicherzuordnungsanforderung, nachdem die Speicherzuordnungsanforderung empfangen worden ist, und erneutes Identifizieren der untätigen SLC-Speicherseite als eine aktive SLC-Speicherseite. 5
7. Verfahren nach Anspruch 6, wobei das erneute Identifizieren der untätigen SLC-Speicherseite als eine aktive SLC-Speicherseite Folgendes umfasst: Übertragen einer Adresse der untätigen SLC-Speicherseite von einer aufgebauten verknüpften Liste untätiger SLC-Speicherseiten zu einer aufgebauten verknüpften Liste aktiver SLC-Speicherseiten. 10
8. Verfahren nach einem der Ansprüche 1 bis 7, wobei das Verfahren ferner Folgendes umfasst: erneutes Identifizieren einer inaktiven SLC-Speicherseite als eine aktive SLC-Speicherseite, nachdem eine Zugriffsanforderung für die inaktive SLC-Speicherseite empfangen worden ist. 15
9. Verfahren nach Anspruch 8, wobei das erneute Identifizieren eine inaktiven SLC-Speicherseite als eine aktive SLC-Speicherseite Folgendes umfasst: Übertragen einer Adresse der inaktiven SLC-Speicherseite von einer aufgebauten verknüpften Liste inaktiver SLC-Speicherseiten zur aufgebauten verknüpften Liste aktiver SLC-Speicherseiten. 20
10. Verfahren nach einem der Ansprüche 1 bis 9, wobei das Verfahren ferner Folgendes umfasst: für eine aktive SLC-Speicherseite, dann, wenn im bestimmten Zeitraum keine Zugriffsanforderung für die Speicherseite empfangen wird, erneutes Identifizieren der aktiven SLC-Speicherseite als eine inaktive SLC-Speicherseite. 25
11. Verfahren nach Anspruch 10, wobei das erneute Identifizieren der aktiven SLC-Speicherseite als eine inaktive SLC-Speicherseite Folgendes umfasst: Übertragen einer Adresse der aktiven SLC-Speicherseite von der aufgebauten verknüpften Liste aktiver SLC-Speicherseiten zu einer aufgebauten verknüpften Liste inaktiver SLC-Speicherseiten. 30
12. Vorrichtung (70), um eine Speicherseite in einem Speicher (72a) zu verarbeiten, **dadurch gekennzeichnet, dass** die Speicherseite im Speicher (72a) eine untätige Speicherseite mit einstufigen Zellen, SLC-Speicherseite, eine aktive SLC-Speicherseite, eine inaktive SLC-Speicherseite und eine Speicherseite mit mehrstufigen Zellen, MLC-Speicherseite, umfasst, wobei die untätige SLC-Speicherseite keine Daten enthält, Daten, die in der aktiven SLC-Speicherseite gespeichert sind, Daten sind, auf die in einem bestimmten Zeitraum zugegriffen wird, und Daten, die in der inaktiven SLC-Speicherseite und der MLC-Speicherseite gespeichert sind, Daten sind, auf die im bestimmten Zeitraum nicht zugegriffen wird; und die Vorrichtung Folgendes umfasst: 35
- ein Umwandlungsmodul (61), das konfiguriert ist, eine untätige SLC-Speicherseite in zwei MLC-Speicherseiten umzuwandeln, wenn bestimmt wird, dass eine Anzahl von untätigen Speicherseiten mit einstufigen Zellen, SLC-Speicherseiten, in einer beliebigen virtuellen Maschine, VM, geringer als ein bestimmter Schwellenwert ist; 40
- ein Kopiermodul (62), das konfiguriert ist, Daten in zwei inaktiven SLC-Speicherseiten zu den zwei umgewandelten MLC-Speicherseiten zu kopieren; und
- ein Freigabemodul (63), das konfiguriert ist, den Speicherplatz der zwei inaktiven SLC-Speicherseiten freizugeben, um zwei untätige SLC-Speicherseiten zu erhalten. 45
13. Vorrichtung nach Anspruch 12, wobei eine höhere Speicherzugriffsfrequenz oder eine höhere Speicherverwendung einen kürzeren bestimmten Zeitraum anzeigt oder eine höhere Speicherzugriffsfrequenz und eine höhere Speicherverwendung einen kürzeren bestimmten Zeitraum anzeigt. 50
14. Vorrichtung nach Anspruch 12 oder 13, wobei ein kleinerer bestimmter Schwellenwert eine höhere Speicherzugriffsleistung und eine höhere Wahrscheinlichkeit des Erzeugens eines Seitenfehlers anzeigt. 55
15. Vorrichtung nach Anspruch 14, wobei das Umwandlungsmodul (61) ferner konfiguriert ist, nach dem Bestimmen, dass eine Anzahl von untätigen SLC-Speicherseiten in einer beliebigen VM geringer als der bestimmte Schwellenwert ist, dann, wenn eine Anzahl von aktuell inaktiven SLC-Speicherseiten geringer als 2 ist, Daten in den zwei MLC-Speicherseiten zurück zu einem externen Speicher zu schreiben und die zwei MLC-Speicherseiten zu einer untätigen SLC-Speicherseite umzuwandeln.
16. Vorrichtung nach Anspruch 15, wobei das Umwandlungsmodul (61) ferner konfiguriert ist, nach dem Bestimmen, dass die Anzahl von untätigen SLC-Speicherseiten in einer beliebigen VM geringer als der bestimmte Schwellenwert ist, dann, wenn eine Anzahl von aktuell inaktiven SLC-Speicherseiten geringer als 2 ist und eine Anzahl von aktuellen MLC-Speicherseiten geringer als 2 ist, Erhalten einer untätigen Speicherseite von einer weiteren VM für die beliebige VM.
17. Vorrichtung nach einem der Ansprüche 12 bis 16, wobei die Vorrichtung ferner Folgendes umfasst:

ein erstes Zuordnungsmodul (64), das konfiguriert ist, eine untätige SLC-Speicherseite für eine Speichertzuoordnungsanforderung zuzuordnen, nachdem die Speichertzuoordnungsanforderung empfangen worden ist, und die untätige SLC-Speicherseite als eine aktive SLC-Speicherseite erneut zu identifizieren.

18. Vorrichtung nach Anspruch 17, wobei das erste Zuordnungsmodul (64) speziell konfiguriert ist, eine Adresse der untätigen SLC-Speicherseite von einer aufgebauten verknüpften Liste untätiger SLC-Speicherseiten zu einer aufgebauten verknüpften Liste aktiver SLC-Speicherseiten zu übertragen.
19. Vorrichtung nach einem der Ansprüche 12 bis 18, wobei die Vorrichtung ferner Folgendes umfasst: ein zweite Zuordnungsmodul (65), das konfiguriert ist, eine inaktive SLC-Speicherseite als eine aktive SLC-Speicherseite erneut zu identifizieren, nachdem eine Zugriffsanforderung für die inaktive SLC-Speicherseite empfangen worden ist.
20. Vorrichtung nach Anspruch 19, wobei das zweite Zuordnungsmodul (65) speziell konfiguriert ist, eine Adresse der inaktiven SLC-Speicherseite von einer aufgebauten verknüpften Liste inaktiver SLC-Speicherseiten zur aufgebauten verknüpften Liste aktiver SLC-Speicherseiten zu übertragen.
21. Vorrichtung nach einem der Ansprüche 12 bis 20, wobei die Vorrichtung ferner Folgendes umfasst: ein drittes Zuordnungsmodul (66), das konfiguriert ist, für eine aktive SLC-Speicherseite, dann, wenn im bestimmten Zeitraum keine Zugriffsanforderung für die Speicherseite empfangen wird, die aktive SLC-Speicherseite als eine inaktive SLC-Speicherseite erneut zu identifizieren.
22. Vorrichtung nach Anspruch 21, wobei das dritte Zuordnungsmodul (66) speziell konfiguriert ist, zum Übertragen einer Adresse der aktiven SLC-Speicherseite von der aufgebauten verknüpften Liste aktiver SLC-Speicherseiten zu einer aufgebauten verknüpften Liste inaktiver SLC-Speicherseiten.

## Revendications

1. Procédé de traitement d'une page de mémoire dans une mémoire (72a), **caractérisé en ce que** la page de mémoire dans la mémoire (72a) comprend une page de mémoire à cellule de niveau unique, SLC, au repos, une page de mémoire SLC active, une page de mémoire SLC inactive, et une page de mémoire à cellule à niveaux multiples, MLC, dans lequel la page de mémoire SLC au repos ne comporte pas de données, les données mémorisées dans la page

de mémoire SLC active sont des données auxquelles on a accédé dans une période de temps spécifiée, et les données mémorisées dans la page de mémoire SLC inactive et la page de mémoire MLC sont des données auxquelles on n'a pas accédé dans la période de temps spécifiée ; et le procédé de traitement comprend :

la conversion (S201) d'une seule page de mémoire SLC au repos en deux pages de mémoire MLC quand il est déterminé qu'une quantité de pages de mémoire à cellule de niveau unique, SLC, au repos dans n'importe quelle machine virtuelle, VM, est inférieure à un seuil spécifié ; la copie (S202) de données se trouvant dans deux pages de mémoire SLC inactives dans les deux pages de mémoire MLC converties ; et la libération (S203) d'un espace de mémorisation des deux pages de mémoire SLC inactives pour obtenir deux pages de mémoire SLC au repos.

2. Procédé selon la revendication 1, dans lequel une fréquence d'accès mémoire supérieure ou une utilisation de mémoire supérieure indique une période de temps spécifiée plus courte ; ou une fréquence d'accès mémoire supérieure et une utilisation de mémoire supérieure indiquent une période de temps spécifiée plus courte.
3. Procédé selon la revendication 1 ou 2, dans lequel un seuil spécifié inférieur indique une performance supérieure d'accès mémoire et une plus grande probabilité de générer une erreur de page.
4. Procédé selon l'une quelconque des revendications 1 à 3, comprenant en outre, après la détermination qu'une quantité de pages de mémoire SLC au repos dans n'importe quelle VM est inférieure à un seuil spécifié : si une quantité de pages de mémoire SLC inactives actuelles est inférieure à 2, la réécriture de données se trouvant dans les deux pages de mémoire MLC dans une mémoire externe, et la conversion des deux pages de mémoire MLC en une seule page de mémoire SLC au repos.
5. Procédé selon la revendication 4, comprenant en outre, après la détermination qu'une quantité de pages de mémoire SLC au repos dans n'importe quelle VM est inférieure à un seuil spécifié : si une quantité de pages de mémoire SLC inactives actuelles est inférieure à 2 et une quantité de pages de mémoire MLC actuelles est inférieure à 2, l'obtention d'une page de mémoire au repos d'une autre VM pour la n'importe quelle VM.
6. Procédé selon l'une quelconque des revendications

- 1 à 5, le procédé comprenant en outre :  
l'allocation d'une page de mémoire SLC au repos à  
une demande d'allocation de mémoire après réception de la demande d'allocation de mémoire, et la  
reclassification de la page de mémoire SLC au repos en tant que page de mémoire SLC active.
7. Procédé selon la revendication 6, dans lequel la reclassification de la page de mémoire SLC au repos en tant que page de mémoire SLC active comprend :  
le transfert d'une adresse de la page de mémoire SLC au repos d'une liste liée de pages de mémoire SLC au repos établie vers une liste liée de pages de mémoire SLC actives établie.
8. Procédé selon l'une quelconque des revendications 1 à 7, le procédé comprenant en outre :  
la reclassification d'une page de mémoire SLC inactive en tant que page de mémoire SLC active après réception d'une demande d'accès à la page de mémoire SLC inactive.
9. Procédé selon la revendication 8, dans lequel la reclassification d'une page de mémoire SLC inactive en tant que page de mémoire SLC active comprend :  
le transfert d'une adresse de la page de mémoire SLC inactive d'une liste liée de pages de mémoire SLC inactives établie vers la liste liée de pages de mémoire SLC actives établie.
10. Procédé selon l'une quelconque des revendications 1 à 9, le procédé comprenant en outre :  
pour une page de mémoire SLC active, si aucune demande d'accès à la page de mémoire n'est reçue dans la période de temps spécifiée, la reclassification de la page de mémoire SLC active en tant que page de mémoire SLC inactive.
11. Procédé selon la revendication 10, dans lequel la reclassification de la page de mémoire SLC active en tant que page de mémoire SLC inactive comprend :  
le transfert d'une adresse de la page de mémoire SLC active de la liste liée de pages de mémoire SLC actives établie vers une liste liée de pages de mémoire SLC inactives établie.
12. Appareil (70) de traitement d'une page de mémoire dans une mémoire (72a), **caractérisé en ce que** la page de mémoire dans la mémoire (72a) comprend une page de mémoire à cellule de niveau unique, SLC, au repos, une page de mémoire SLC active, une page de mémoire SLC inactive, et une page de mémoire à cellule à niveaux multiples, MLC, dans lequel la page de mémoire SLC au repos ne comporte pas de données, les données mémorisées dans la page de mémoire SLC active sont des données auxquelles on a accédé dans une période de temps spécifiée, et les données mémorisées dans la page de mémoire SLC inactive et la page de mémoire MLC sont des données auxquelles on n'a pas accédé dans la période de temps spécifiée ; et l'appareil comprend :  
un module de conversion (61), configuré pour convertir une seule page de mémoire SLC au repos en deux pages de mémoire MLC quand il est déterminé qu'une quantité de pages de mémoire à cellule de niveau unique, SLC, au repos dans n'importe quelle machine virtuelle, VM, est inférieure à un seuil spécifié ;  
un module de copie (62), configuré pour copier des données se trouvant dans deux pages de mémoire SLC inactives dans les deux pages de mémoire MLC converties ; et  
un module de libération (63), configuré pour libérer un espace de mémorisation des deux pages de mémoire SLC inactives pour obtenir deux pages de mémoire SLC au repos.
13. Appareil selon la revendication 12, dans lequel une fréquence d'accès mémoire supérieure ou une utilisation de mémoire supérieure indique une période de temps spécifiée plus courte ; ou une fréquence d'accès mémoire supérieure et une utilisation de mémoire supérieure indiquent une période de temps spécifiée plus courte.
14. Appareil selon la revendication 12 ou 13, dans lequel un seuil spécifié inférieur indique une performance supérieure d'accès mémoire et une plus grande probabilité de générer une erreur de page.
15. Appareil selon la revendication 14, dans lequel le module de conversion (61) est configuré en outre pour, après la détermination qu'une quantité de pages de mémoire SLC au repos dans n'importe quelle VM est inférieure au seuil spécifié, si une quantité de pages de mémoire SLC inactives actuelles est inférieure à 2, réécrire des données se trouvant dans les deux pages de mémoire MLC dans une mémoire externe, et convertir les deux pages de mémoire MLC en une seule page de mémoire SLC au repos.
16. Appareil selon la revendication 15, dans lequel le module de conversion (61) est configuré en outre pour, après la détermination que la quantité de pages de mémoire SLC au repos dans n'importe quelle VM est inférieure à un seuil spécifié, si une quantité de pages de mémoire SLC inactives actuelles est inférieure à 2 et une quantité de pages de mémoire MLC actuelles est inférieure à 2, obtenir une page de mémoire au repos d'une autre VM pour la n'importe quelle VM.
17. Appareil selon l'une quelconque des revendications

12 à 16, l'appareil comprenant en outre :

un premier module d'allocation (64), configuré pour allouer une page de mémoire SLC au repos à une demande d'allocation de mémoire après réception de la demande d'allocation de mémoire, et reclassifier la page de mémoire SLC au repos en tant que page de mémoire SLC active.

5

18. Appareil selon la revendication 17, dans lequel le premier module d'allocation (64) est configuré spécifiquement pour transférer une adresse de la page de mémoire SLC au repos d'une liste liée de pages de mémoire SLC au repos établie vers la liste liée de pages de mémoire SLC actives établie.

10

15

19. Appareil selon l'une quelconque des revendications 12 à 18, l'appareil comprenant en outre : un deuxième module d'allocation (65), configuré pour reclassifier une page de mémoire SLC inactive en tant que page de mémoire SLC active après réception d'une demande d'accès à la page de mémoire SLC inactive.

20

20. Appareil selon la revendication 19, dans lequel le deuxième module d'allocation (65) est configuré spécifiquement pour transférer une adresse de la page de mémoire SLC inactive d'une liste liée de pages de mémoire SLC inactives établie vers la liste liée de pages de mémoire SLC actives établie.

25

30

21. Appareil selon l'une quelconque des revendications 12 à 20, l'appareil comprenant en outre : un troisième module d'allocation (66), configuré pour, pour une page de mémoire SLC active, si aucune demande d'accès à la page de mémoire n'est reçue dans la période de temps spécifiée, reclassifier la page de mémoire SLC active en tant que page de mémoire SLC inactive.

35

22. Appareil selon la revendication 21, dans lequel le troisième module d'allocation (66) est configuré spécifiquement pour : transférer une adresse de la page de mémoire SLC active de la liste liée de pages de mémoire SLC actives établie vers une liste liée de pages de mémoire SLC inactives établie.

40

45

50

55

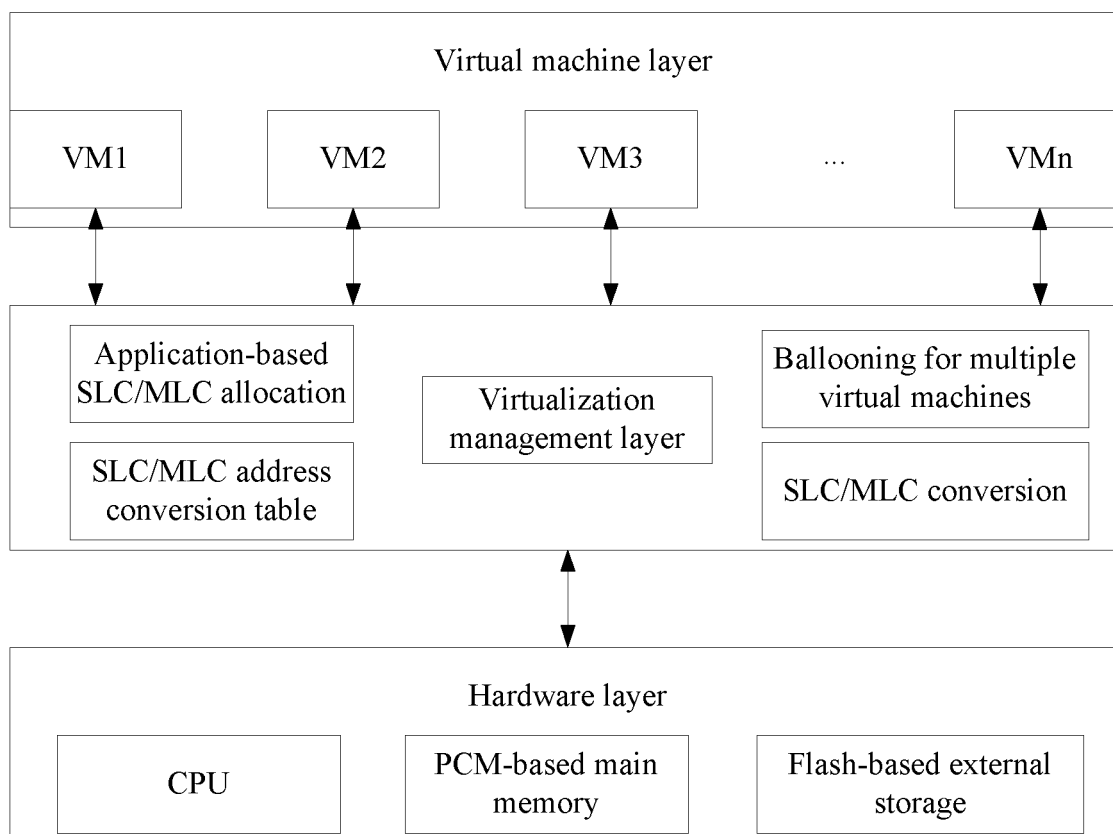


FIG. 1

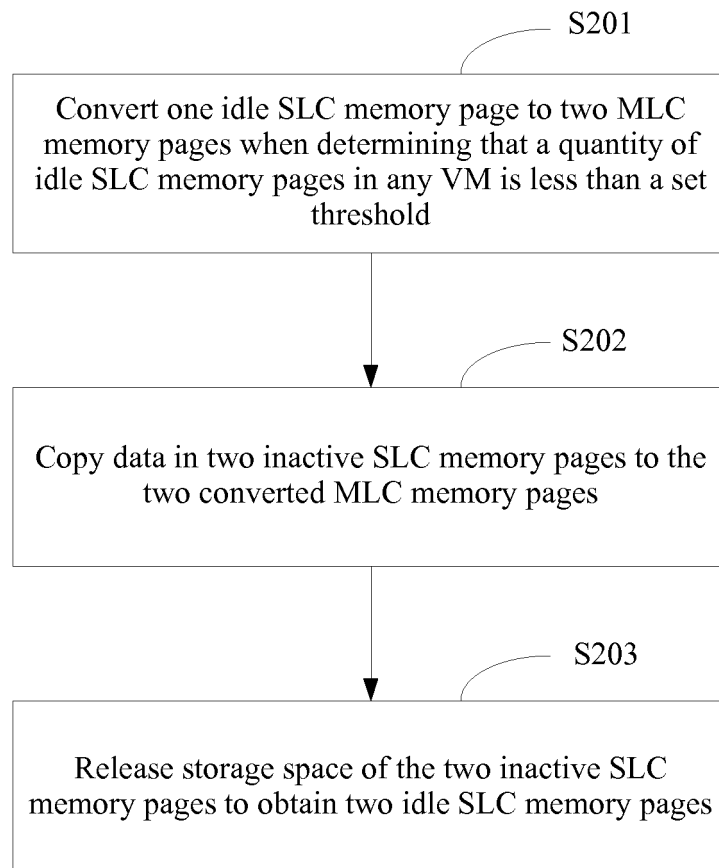


FIG. 2

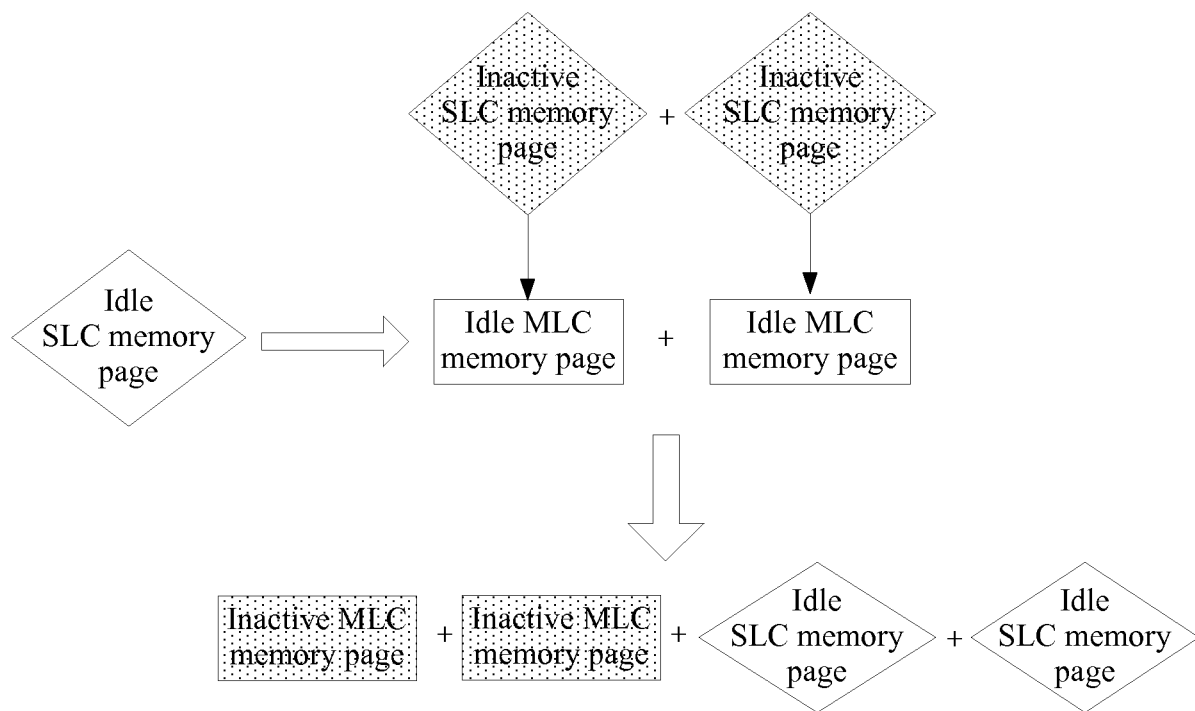


FIG. 3

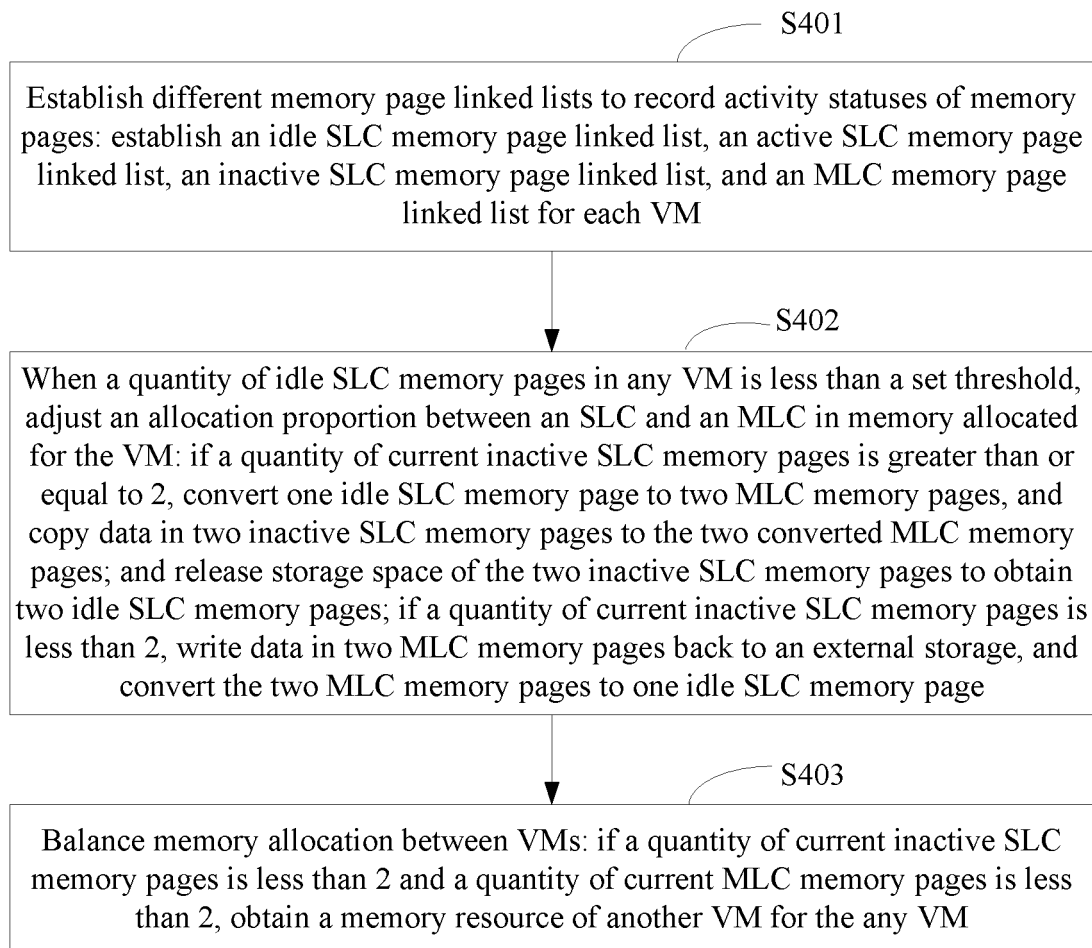


FIG. 4

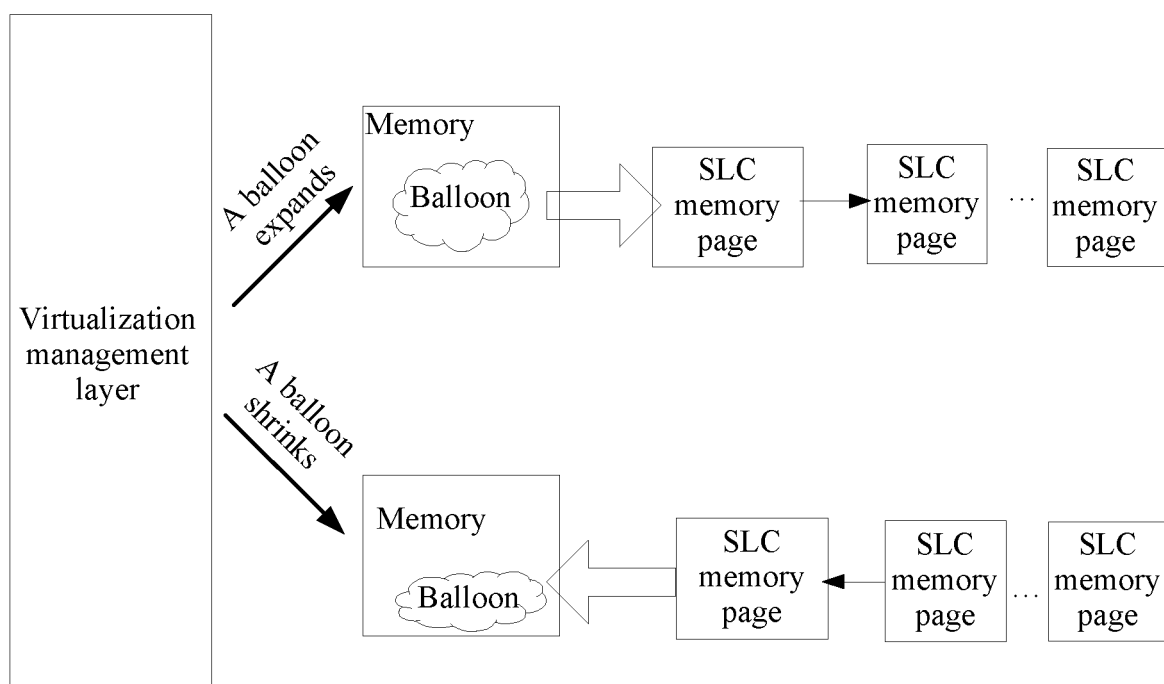


FIG. 5

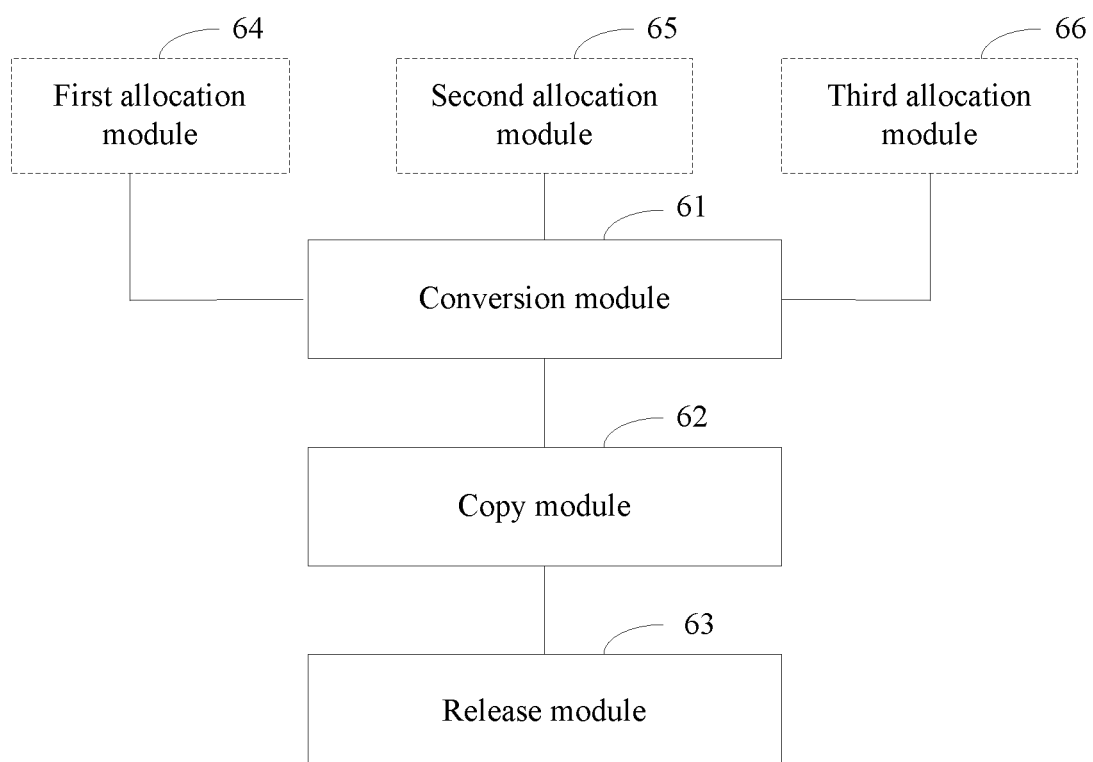


FIG. 6

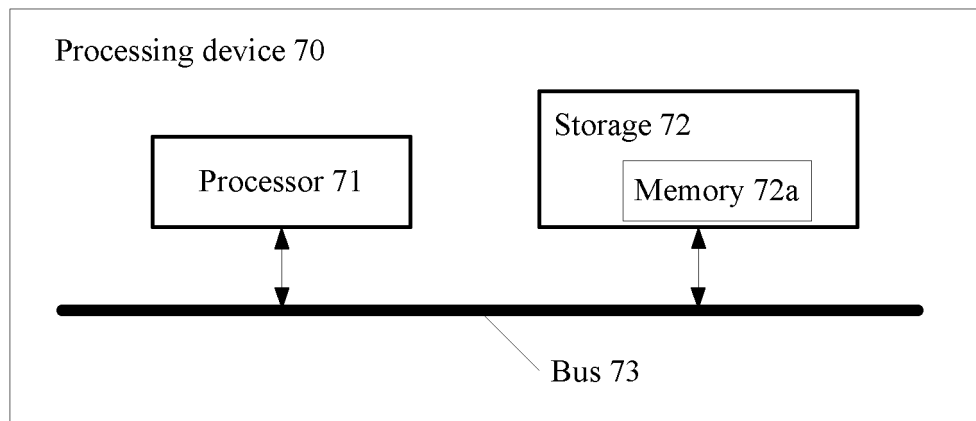


FIG. 7

**REFERENCES CITED IN THE DESCRIPTION**

*This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.*

**Patent documents cited in the description**

- US 2012030405 A1 [0005]
- US 2014003142 A1 [0006]